



# HPC Resilience Kaizen: Metrics and Modeling

**2009 National HPC Workshop on Resilience  
August 13 2009, Washington DC**

**Jon Stearley**

**[jrstear@sandia.gov](mailto:jrstear@sandia.gov)**

**Sandia National Laboratories**



Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,  
for the United States Department of Energy's National Nuclear Security Administration  
under contract DE-AC04-94AL85000.



**If you can not measure it,  
you can not improve it.**

**- Lord Kelvin**

## **What is Resilience?**

**We must learn to:**

- 1. quantify it consistently (metrics).**
- 2. understand it at a system-level (modeling).**



# Not resilience, but related...

## Downtime:

**Expensive!**

**Minimize it!**

Semiconductor Manufacturing:

SEMI-E10

XSite

**We must learn to:**

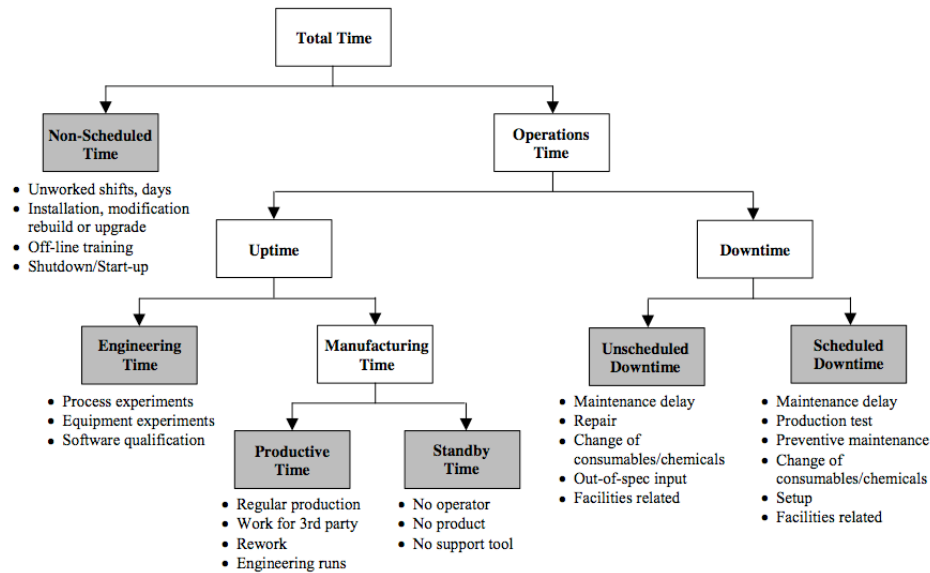
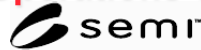
- 1. quantify it consistently (metrics).**
- 2. understand it at a system-level (modeling).**



# Semiconductor Manufacturing

## SEMI-E10 (Metrics)

### Operations Status



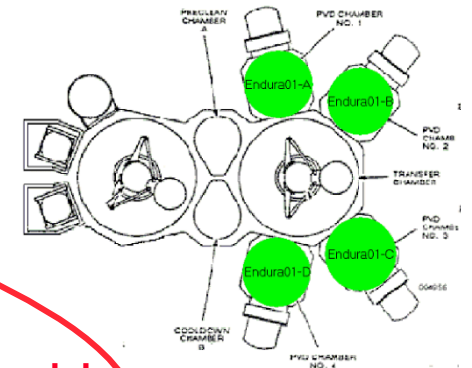
### Metrics

6.3 **EQUIPMENT AVAILABILITY** — The probability that the equipment will be in a condition to perform its intended function when required.

6.3.1 **Equipment Dependent Uptime** — The percent of time the equipment is in a condition to perform its intended function during the period of operations time minus the sum of all maintenance delay downtime, out-of-spec input downtime, and facilities-related downtime. This calculation is intended to reflect equipment reliability and maintainability based solely on equipment merit.

$$\text{equipment dependent uptime (\%)} = \frac{\text{equipment uptime} \times 100}{\text{oper-time} - (\text{all maint-delay DT} + \text{out-of-spec input DT} + \text{fac-rel DT})}$$

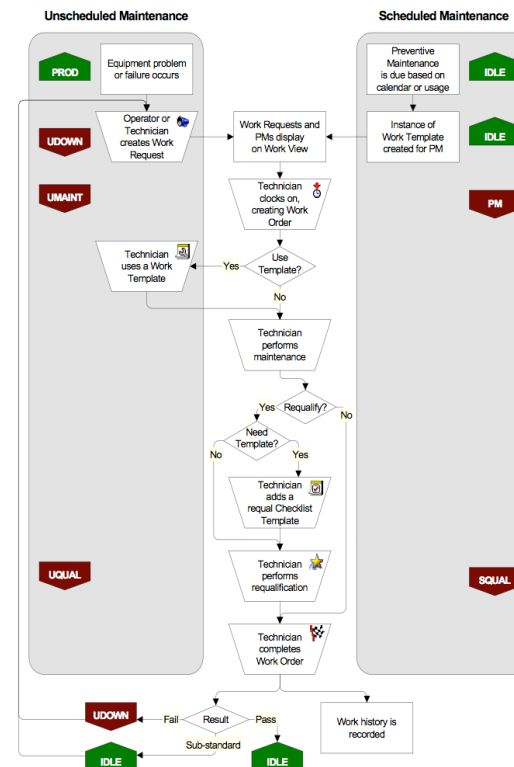
## Xsite (Modeling)



### Physical model

### Monitoring

### Logical model



### Operations Status

### Metrics

### Improvements

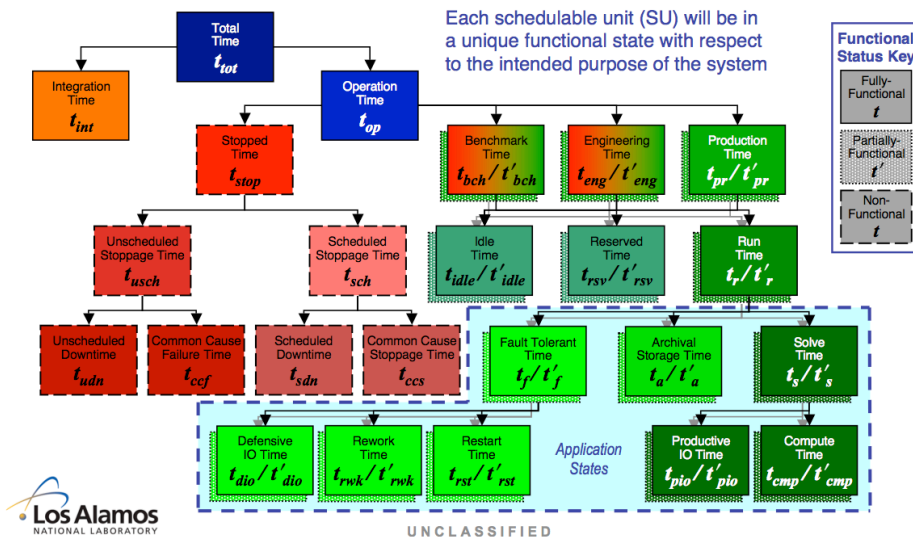


# We need the same for HPC!

## Metrics

### Operations Status

#### Proposed System State Diagram (version 14)

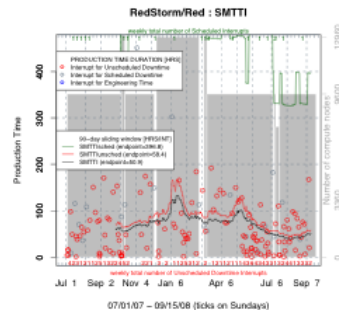


### Metrics

We define a system interrupt as *any* interruption in Production Up-time. The system mean time to interrupt (SMTTI) is calculated as the total number of hours spent in Production Up-time, divided by the total number of system interrupts. All metrics described herein are calculated using a 90-day sliding window.

$$SMTTI = \frac{\text{ProductionUpTime}}{\text{SystemInterrupts}} \left[ \frac{\text{HRS}}{\text{INT}} \right] \quad (1)$$

When this metric is calculated using only scheduled interrupts, we refer to it as SMTTIsched. When it is calculated using only unscheduled interrupts, we refer to it as SMTTIunsched. These metrics are tracked for the Red and Black sides. The per-side plots at right also show the duration of Production Up-time periods (circles), and the number of compute nodes present (grey background).



## Modeling

Physical model

Monitoring

Logical model

#### Proposed Tuples Diagram (version 3b)

##### Operations Status

|            |             |           |                    |                      |         |
|------------|-------------|-----------|--------------------|----------------------|---------|
| Production | Engineering | Benchmark | Scheduled Stoppage | Unscheduled Stoppage | Unknown |
|------------|-------------|-----------|--------------------|----------------------|---------|

##### Job Status

|     |          |      |         |
|-----|----------|------|---------|
| Run | Reserved | Idle | Unknown |
|-----|----------|------|---------|

##### Functional Status

|                  |                    |                |         |
|------------------|--------------------|----------------|---------|
| Fully-Functional | Partial (Degraded) | Non-Functional | Unknown |
|------------------|--------------------|----------------|---------|

##### Dependency Status

|                   |              |         |
|-------------------|--------------|---------|
| Local Independent | Common Cause | Unknown |
|-------------------|--------------|---------|

Operations Status

Metrics

Improvements



# Metrics

## Status

**There is firm consensus that current metrics are inconsistent:**

- Site vs Site [1,2]
- Application vs System [3]

**For \$100M+ systems, and the science that depends on them, we must do better than “practically perfect in every way!”**

**Gordon Bell (et al) provides excellent guidance, but implementation will require better **operations status** data [4].**

- **Production Uptime**
- **Scheduled Downtime**
- **Unscheduled Downtime**

## Needed

**1. Publish a multi-agency standard for resilience metrics.**

**2. Provide a reference implementation.**

**3. Require compliance on all future systems.**

[1] Stearley. Defining and Measuring Supercomputer Reliability, Availability, and Serviceability. LCI Conference. 2005

[2] TriPOD metrics group [Cupps, Rheinheimer, Stearley, et al]

[3] Daly. Performance Challenges for Extreme Scale Computing, SDI/LCS Seminar. 2007. (unavailability) “may differ by as much as a factor of three”

[4] Bell, Hack, et al. Advanced Scientific Computing Advisory Committee Petascale Metrics Report. 2007.



# (real system) Modeling

## Status

Subsystems developed too independently  
(eg I/O vs compute vs workload)  
I/O is a primary system weakness,  
but is also the sole fault mitigation strategy  
(checkpoint!)

Various cabling diagrams

Distributed configuration files

Distributed sensor info (logs, numeric)

Mental models (tribal knowledge)

High integration and debugging costs

## Future

Higher component counts

Higher complexity

## Needed

### 1.Full-system models

- Integrated physical, logical, workload, and sensor info
- Understanding of interdependencies
- **Operations status** data

### 2.New approaches to faults:

- **Computation** of root cause and total effect
- **Coordinated** response (eg application and subsystems [1,2])

### 3.Productive feedback loop among researchers, operators, and agencies based on accurate operations status data

[1] Oliner. Cooperative Checkpointing for Supercomputing Systems. Thesis. 2005

[2] CIFS: A Coordinated Infrastructure for Fault-Tolerant Systems. 2009



**If you can not measure it,  
you can not improve it.**

**- Lord Kelvin**

## **What is Resilience?**

**We need a unifying catalyst to:**

- 1. Establish standard metrics.**
- 2. Develop full-system models.**

eg:

E10

XSite



**Extra slides...**



# Modeling

**Needed**

**Independent** components VS **Interdependent** components  
*(the latter is what counts)*

**A true system-level model will include:**

- **physical info (cabinets, boards, cables, ...)**
- **logical info (routing, node role, software versions, ...)**
- **workload info (job, application, filesystem, ...)**
- **downtime info (scheduled, unscheduled, reason)**

**No full-system models or tools exists;**

**Our system-level understanding is insufficient.**

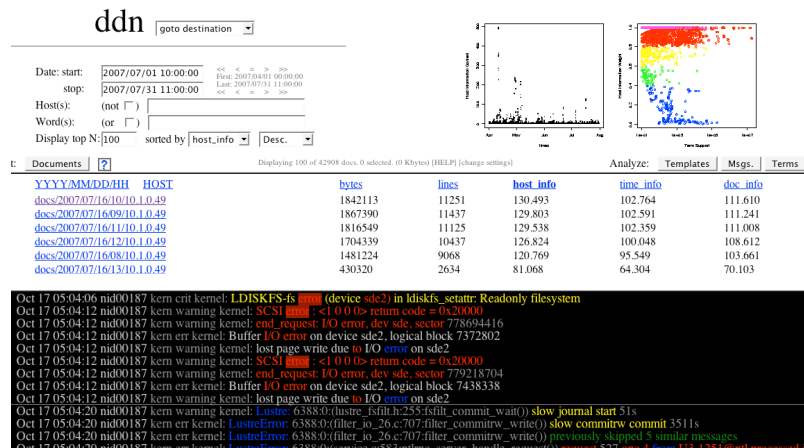


# My Background

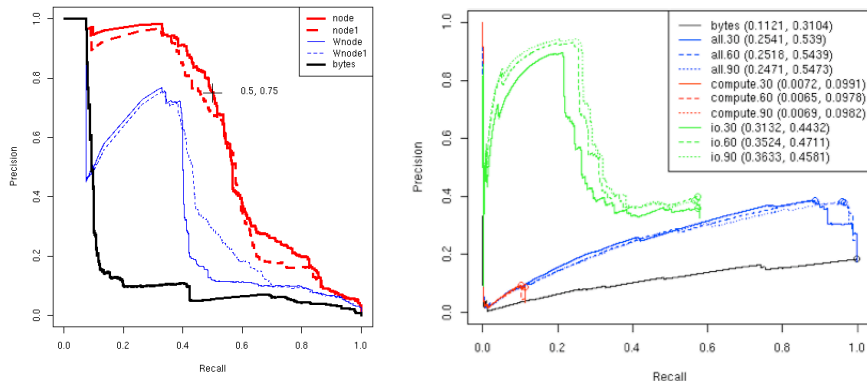


## Data mining of system logs

### Intuitive tools



### Quantitative, validated results



<http://www.cs.sandia.gov/sisyphus>  
<http://cfdr.usenix.org/data.html#hpc4>

## RAS Metrics

### RAS = Reliability, Availability, Serviceability

#### Mathematical Formulation of the Red Storm 100Hr SOW Criteria

"MTBI for the full system, as determined by the need to reboot the system, shall be greater than 100 hours of continuous operation. This means that the system will be continuously operational for 100 hours with at least 99% of the system resources available and all disk storage accessible."

Based on incident report emails from the operations staff, Sandia tracks the Red and Black sides of the Red Storm system as being in exactly one of three states at all times:



We define a system interrupt as any interruption in Production Uptime. The system mean time to interrupt (SMTTI) is calculated as the total number of hours spent in Production Uptime, divided by the total number of system interrupts. All metrics described herein are calculated using a 90-day sliding window.

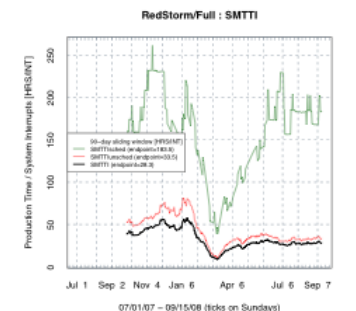
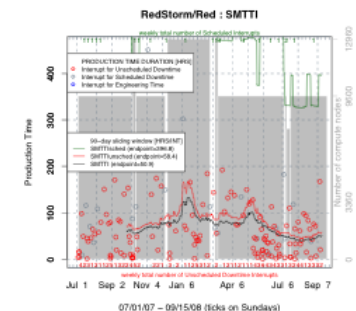
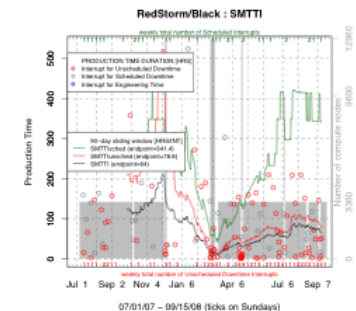
$$SMTTI = \frac{ProductionUptime}{SystemInterrupts} [HRS/INT] \quad (1)$$

When this metric is calculated using only scheduled interrupts, we refer to it as SMTTIsched. When it is calculated using only unscheduled interrupts, we refer to it as SMTTIunsched. These metrics are tracked for the Red and Black sides. The per-side plots at right also show the duration of Production Uptime periods (circles), and the number of compute nodes present (grey background).

In order to measure the 100Hr criteria for the full system, the Red and Black sides are modeled as being in series. The failure rate (INT/Hr) of a series system is equal to the sum of the failure rates of the components, and it follows that:

$$\frac{1}{fullSMTTI} = \frac{1}{redSMTTI} + \frac{1}{blackSMTTI} \quad (2)$$

The above formulation for fullSMTTI is being considered for use as Sandia's official measurement of the 100Hr SOW criteria (see bold black line at right).



<http://www.cs.sandia.gov/~jrstear/ras>



# Idea : Compute Cause and Effect

## 1. Given a System

### Components

- Hardware (racks, cores, routes, ...)
- Software (daemons, apps, libraries, ...)

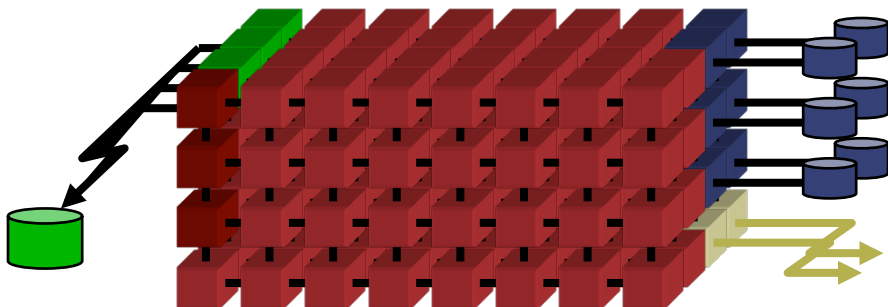
### Dependencies

- Logical, Physical
- Known, Inferred

### Observations

- Text (logs, version numbers, username, application name, rank id, ...)
- Numbers (counters, sensors, correlation, ...)

(all the above are dynamic and incomplete)

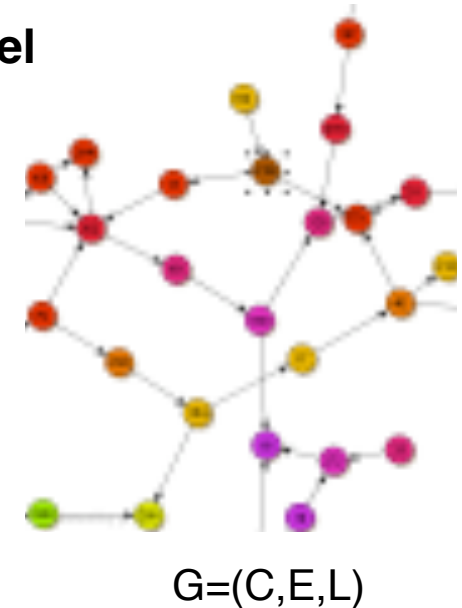


## 2. Construct a Model

VERTICES (C)

EDGES (E)

LABELS (L)



## 3. Compute Cause and Effect

**Input:** suspect C, E, or L

**Output:** likelihood-ranked causes and system-wide effects.

13K computers (<10 changes/month)  
in 7 functional roles (<5 changes/month)  
eg 320 RAID servers, 8 disks/RAID  
88M pair-wise routes (15 reroutes/month)  
500 users, 50 applications (5k jobs/month)



Excerpt from Bell, Hack, et al.  
**Advanced Scientific Computing Advisory Committee Petascale Metrics  
Report. 2007**



1.2: Availability- Systems are available to process a workload. Meeting the availability metric means the machines are up and available nearly all of the time. Scheduled availability targets should be determined per-machine, based on the capabilities, characteristics, and mission of that machine. Availabilities are of interest both at the initial startup to understand the time to reach a stable operational state and later in the machine lifetime to understand failures.

The Panel recommends that scheduled availability be a control metric, where scheduled availability is the percentage of time a system is available for users, accounting for any scheduled downtime for maintenance and upgrades.

$$= (\Sigma \text{ scheduled hours} - \Sigma \text{ outages during scheduled time}) / \Sigma \text{ scheduled hours}$$

A service interruption is any event or failure (hardware, software, human, and environment) that **degrades service below an agreed-upon threshold**. With modern scalable computers, the threshold will be system dependent; where the idea is that the failure of just a few nodes in a multi-thousand node machine need not constitute a service interruption. Any shutdown that has **less than 24 hours notice is treated as an unscheduled interruption**. A service outage is the time from when computational processing halts to the **restoration of full operational capability** (e.g., not when the system was booted, but rather when user jobs are recovered and restarted). The centers should be expected to demonstrate that **within 12 months of delivery**, or a suitable period following a significant upgrade, scheduled availability is >95% or another value agreed to by ASCR.

The Panel recommends that overall availability be an observed metric, where overall availability is the percentage of time a system is available for users, based on the total time of the period.

$$= (\Sigma \text{ Total clock hours} - \Sigma (\text{outages, upgrades, scheduled maintenance, etc.})) / \Sigma \text{ Total clock hours}$$

Using overall availability as a control metric may easily become counter productive as it can inhibit beneficial upgrades.